A Union Bound Approximation for Rapid Performance Evaluation of Punctured Turbo Codes

Ioannis Chatzigeorgiou, Miguel R. D. Rodrigues, Ian J. Wassell
Digital Technology Group, Computer Laboratory
University of Cambridge, United Kingdom
Email: {ic231, mrdr3, ijw24}@cam.ac.uk

Rolando Carrasco
School of EE&C Engineering
University of Newcastle, United Kingdom
Email: r.carrasco@ncl.ac.uk

Abstract—In this paper, we present a simple technique to approximate the performance union bound of a punctured turbo code. The bound approximation exploits only those terms of the transfer function that have a major impact on the overall performance. We revisit the structure of the constituent convolutional encoder and we develop a rapid method to calculate the most significant terms of the transfer function of a turbo encoder. We demonstrate that, for a large interleaver size, this approximation is very accurate. Furthermore, we apply our proposed method to a family of punctured turbo codes, which we call pseudo-randomly punctured codes. We conclude by emphasizing the benefits of our approach compared to those employed previously. We also highlight the advantages of pseudo-random puncturing over other puncturing schemes.

I. INTRODUCTION

Turbo codes, originally conceived by Berrou *et al.* [1] are widely known for their astonishing performance on the additive white Gaussian noise (AWGN) channel. Methods to evaluate an upper bound on the bit error probability (BEP) of a parallel-concatenated coding scheme have been proposed by Divsalar *et al.* [2] as well as Benedetto and Montorsi [3]. In addition, guidelines for the optimal design of the constituent convolutional codes were presented in [4].

The rate of a turbo code can be increased by puncturing the outputs of the turbo encoder. Guidelines and design considerations for punctured turbo codes have been derived by analytical [5]–[7] as well as simulation-based approaches [8], [9]. Upper bounds on the bit error probability (BEP) can be easily evaluated based on the techniques presented in [7] and [10]. However, computation of the upper bound can be complex and time-consuming, when a large interleaver size and certain puncturing patterns are used.

The motivation for this paper is to derive simple expressions for the calculation of the dominant term of the performance union bound for punctured parallel concatenated convolutional codes (PCCCs). Previously, complex approaches based on the full transfer function of each constituent code, have been used. In Section II we demonstrate that for a large interleaver size, the dominant term can be used as an accurate approximation of the overall performance union bound. In Section III we analyze the properties of constituent convolutional encoders so as to obtain exact expressions for the dominant term. A case

study considering pseudo-random puncturing is presented in Section IV and the paper concludes with a summary of the main contributions.

II. AN UPPER BOUND TO THE ERROR PROBABILITY OF PUNCTURED TURBO CODES AND ITS APPROXIMATION

Turbo codes, in the form of rate-1/3 PCCCs, consist of two rate-1/2 recursive systematic convolutional (RSC) encoders separated by an interleaver of size N [1]. The information bits are input to the first constituent RSC encoder, while an interleaved version of the information bits are input to the second RSC encoder. The output of the turbo encoder consists of the systematic bits of the first encoder, which are identical to the information bits, the parity-check bits of the first encoder and the parity-check bits of the second encoder.

Rates higher than 1/3 can be obtained by periodic elimination of specific codeword bits from the output of a rate-1/3 turbo encoder. Punctured codes are classified as systematic (S), partially systematic (PS) or non-systematic (NS) depending on whether all, some or none of their systematic bits are transmitted [9]. Note that a punctured PCCC can also be seen as a PCCC constructed using two constituent punctured RSC codes.

Puncturing of a rate-1/2 RSC to obtain a higher rate RSC is represented by an $2 \times M$ matrix as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_U \\ \mathbf{P}_Z \end{bmatrix} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,M} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,M} \end{bmatrix}, \qquad (1)$$

where M is the puncturing period and $p_{i,m} \in \{0,1\}$, with $i = 1, 2$ and $m = 1, \ldots, M$. For $p_{i,m} = 0$, the corresponding output bit is punctured. The puncturing pattern $\mathbf{P}$ for the rate-1/2 encoder consists of the puncturing vector $\mathbf{P}_U$ for the systematic output sequence and the puncturing vector $\mathbf{P}_Z$ for the parity-check output sequence.

It was shown in [2] and [3] that performance bounds for a PCCC can be obtained from the transfer functions, or equivalently the weight enumerating functions (WEFs), of the terminated constituent RSC codes. A WEF provides all paths of length N that start from the zero state, can remerge with and diverge from the zero state more than once, and terminate at the zero state.

More specifically, the conditional WEF (CWEF) of a punctured convolutional code $\mathcal{C}'$, denoted as $A^{\mathcal{C}'}(w, U, Z)$,

assumes the form [3]

$$A^{\mathcal{C}'}(w, U, Z) = \sum_u \sum_z A^{\mathcal{C}'}_{w,u,z} U^u Z^z, \qquad (2)$$

where $A^{\mathcal{C}'}_{w,u,z}$ is the number of codeword sequences composed of a systematic and a parity-check sequence having weights u and z, respectively, which were generated by input sequences of a given weight w. The overall weight of a codeword sequence is $u + z$.

The input-redundancy WEF (IRWEF), $A^{\mathcal{C}'}(W, U, Z)$, provides all codeword sequences for all possible values of input information weight, and is related to the CWEF as follows [3]

$$A^{\mathcal{C}'}(W, U, Z) = \sum_w A^{\mathcal{C}'}(w, U, Z) W^w. \qquad (3)$$

A relationship between the CWEF of a PCCC and the CWEFs of the constituent codes, $\mathcal{C}'_1$ and $\mathcal{C}'_2$ respectively, can be easily derived only if we assume the use of a uniform interleaver of size N, an abstract probabilistic concept introduced in [3]. More specifically, if $A^{\mathcal{C}'_1}(w, U, Z)$ and $A^{\mathcal{C}'_2}(w, U, Z)$ are the CWEFs of the constituent codes, the CWEF of the PCCC, $A(w, U, Z)$, is equal to

$$A(w, U, Z) = \frac{A^{\mathcal{C}'_1}(w, U, Z) \cdot A^{\mathcal{C}'_2}(w, U = 1, Z)}{\binom{N}{w}}. \qquad (4)$$

The systematic output sequence of the second constituent encoder is not transmitted, therefore it does not contribute to the overall weight of the turbo codeword sequences, so it is eliminated by setting $U = 1$ in $A^{\mathcal{C}'_2}(w, U, Z)$. The IRWEF of the PCCC, $A(W, U, Z)$, can be computed from the CWEF, $A(w, U, Z)$, in a manner identical to (3).

The input-output weight enumerating function (IOWEF) provides the number of codewords generated by an input sequence of information weight w, whose overall weight is d, in contrast with the IRWEF, which distinguishes between the systematic and the parity-check weights. For the case of a punctured PCCC, the corresponding IOWEF assumes the form

$$B(W, D) = \sum_w \sum_d B_{w,d} W^w D^d, \qquad (5)$$

where the coefficients $B_{w,d}$ can be derived from the coefficients $A_{w,u,z}$ of the IRWEF, based on the expression

$$B_{w,d} = \sum_{u+z=d} A_{w,u,z}. \qquad (6)$$

The IOWEF coefficients $B_{w,d}$ can be used to determine a tight upper bound, denoted as P_b^u, on the BEP P_b, for maximum-likelihood (ML) soft decoding for the case of an AWGN channel, as follows [3]

$$P_b \le P_b^u = \sum_w P(w), \qquad (7)$$

where $P(w)$ is the union bound of all error events with information weight w, and is defined as

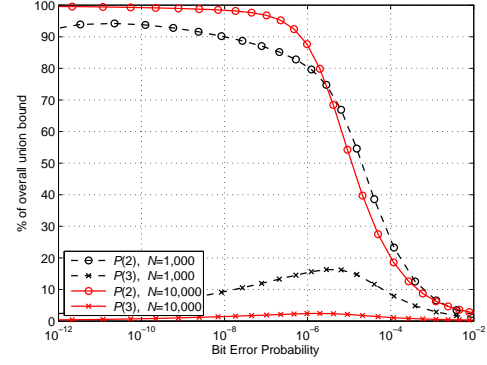$$P(w) = \sum_d \frac{w}{N} B_{w,d} Q\left(\sqrt{\frac{2R \cdot E_b}{N_0} \cdot d}\right), \qquad (8)$$

Fig. 1. Contribution to the union bound for rate-1/2 S-PCCC(1,17/15,17/15) using an interleaver of size $N = 1,000$ and $N = 10,000$ bits

where R is the rate of the punctured turbo code.

In [4], Benedetto *et al.* investigated the performance of rate-1/3 PCCCs and observed that the union bound $P(w_{\min})$ of all error events with the lowest information weight $w_{\min}$, becomes dominant as the interleaver size N increases. Owing to the structure of an RSC encoder, the minimum information weight of a terminated RSC code is always equal to two, i.e., $w_{\min} = 2$. Consequently, the overall performance bound P_b^u can be approximated by $P(2)$, when a large interleaver size is used. The same trend is also observed in the case of punctured turbo codes. The contribution, as a percentage, of $P(2)$ and $P(3)$ to P_b^u is illustrated in Fig.1. As an example, rate-1/2 S-PCCC(1, 17/15, 17/15) is considered, using a uniform interleaver of size either $N = 1,000$ or $N = 10,000$. It is apparent that $P(2)$ becomes the dominant contribution over a broad range of BEP values, as the interleaver size increases.

We see from (8) that $P(2)$ depends heavily on the minimum weight of the turbo codeword sequences, commonly known as free effective distance $d_{\text{free,eff}}$ [4]. We use the notation $d_{\min}^{\mathcal{C}'_1}$ to denote the minimum weight of the codeword sequences generated by the first constituent encoder and $z_{\min}^{\mathcal{C}'_2}$ to denote the minimum weight of the parity-check output sequences, generated by the second constituent encoder. In both cases an input sequence of information weight 2 is assumed. Therefore, the free effective distance of a PCCC can be expressed as

$$d_{\text{free,eff}} = d_{\min}^{\mathcal{C}'_1} + z_{\min}^{\mathcal{C}'_2}. \qquad (9)$$

The free effective distance is the most significant parameter that influences the PCCC performance. The constituent encoders should be chosen to maximise $d_{\min}^{\mathcal{C}'_1}$ and $z_{\min}^{\mathcal{C}'_2}$, and consequently $d_{\text{free,eff}}$.

III. COMPUTING THE UPPER BOUND APPROXIMATION

In order to compute $P(2)$ and thus obtain a good approximation to the overall performance bound P_b^u, we only need to calculate the CWEF of each constituent code for $w = 2$, i.e., $A^{\mathcal{C}'_1}(2, U, Z)$ and $A^{\mathcal{C}'_2}(2, U, Z)$.

Both CWEFs could be obtained by brute-force, i.e., input all possible sequences of weight 2 to each constituent encoder

and group the output codeword sequences according to their systematic and parity-check weights. Although this approach is conceptually simple, it is extremely time-consuming, especially when a large interleaver size is used.

The techniques proposed in [7] and [10] are more complex but less time-consuming. They both use the state diagram of a parent RSC code and introduce the puncturing patterns to obtain the full CWEF of the corresponding punctured RSC code. However, for large interleaver sizes and puncturing patterns with a long period, complexity becomes a prohibiting factor for the implementation of either approach.

In this section we use the properties of the trellis structure of RSC codes to express the CWEF, for $w = 2$, of an RSC encoder as a function of its memory size, generator polynomials, and puncturing pattern. Consequently, derivation of the state equations and computation of the full transfer function of each constituent code, required in [7] and [10], is not necessary. Hence, PCCCs using both a large interleaver and a long puncturing pattern can now be easily supported.

A. Unpunctured Rate-1/2 RSC Encoders

A rate-1/2 RSC encoder, $\mathcal{C}$, is characterised by its feedback and feedforward polynomials, $G_R(D)$ and $G_F(D)$ respectively. The degree of each polynomial is equal to the memory size ν of the encoder. A hypothesis commonly made [1], [4] so as to facilitate analysis of RSC codes is that $G_F(D)$ is a monic function and that the initial state of the encoder is the zero state, for every input sequence of length N.

Input sequences of weight 2 force the trellis path to diverge from the zero state and re-merge with it, after a number of time-steps. More specifically, the input sequence will change the state from 0 to $2^{\nu-1}$, when the first non-zero bit is input to the encoder, as it is illustrated in Fig.2. For as long as a trail of zeros follows the first non-zero input bit, the RSC encoder behaves like a pseudo-random generator, with the same state transitions being repeated every L time-steps, where L is the period of the feedback polynomial. In order for the path to re-merge with the zero state, the second non-zero bit should be input to the encoder when state 1 is reached, i.e., after $kL+1$ time-steps, where $k = 1, 2, \ldots, \lfloor(N-1)/L\rfloor$ and $\lfloor(N-1)/L\rfloor$ is the integer part of $(N-1)/L$. Furthermore, as it is depicted in Fig.2, when a non-zero input bit causes the path to diverge from or re-merge to the zero state, both the systematic and the parity-check outputs give a logical 1. Therefore, if $z_{\text{core}}^{\mathcal{C}}$ is the parity-check weight due to the transitions of the encoder from state $2^{\nu-1}$ to state 1, the overall weight z of a parity-check sequence can be expressed as

$$z(k) = kz_{\text{core}}^{\mathcal{C}} + 2, \quad \text{for } k = 1, 2, \ldots, \lfloor(N-1)/L\rfloor. \quad (10)$$

Note that the state sequence during the transitions from state $2^{\nu-1}$ to state 1 and, consequently, the value of $z_{\text{core}}^{\mathcal{C}}$, depend on the selected feedback polynomial. The minimum parity-check weight $z_{\text{min}}^{\mathcal{C}}$ can be derived from $z(k)$ by setting $k = 1$, i.e.,

$$z_{\text{min}}^{\mathcal{C}} = z(1). \quad (11)$$

Based on (2) and (10), the CWEF for $w = 2$, $A^{\mathcal{C}}(2, U, Z)$, of the rate-1/2 RSC code when no puncturing is applied, assumes the form

$$A^{\mathcal{C}}(2, U, Z) = \sum_{k=1}^{\lfloor(N-1)/L\rfloor} A_{2,2,z(k)}^{\mathcal{C}} U^2 Z^{z(k)}, \quad (12)$$

where $A_{2,2,z(k)}^{\mathcal{C}}$ is the number of codeword sequences with parity-check weight $z(k)$, given by

$$A_{2,2,z(k)}^{\mathcal{C}} = N - kL. \quad (13)$$

When the feedback polynomial, $G_R(D)$, of an RSC encoder is selected to be primitive, the encoder visits all possible $2^{\nu} - 1$ states with a maximum period of $L = 2^{\nu} - 1$ time-steps [11], if the information weight of the input sequence is 2. As pointed out in [12], maximization of L increases the length of the shortest weight 2 input sequence, therefore increasing the chance of achieving a high weight $z_{\text{core}}^{\mathcal{C}}$ and, consequently, $z_{\text{min}}^{\mathcal{C}}$. An exact expression for $z_{\text{core}}^{\mathcal{C}}$ can be derived based on the properties of pseudo-random sequences [11] or the analysis in [4], i.e.,

$$z_{\text{core}}^{\mathcal{C}} = 2^{\nu-1}, \quad (14)$$

provided that $G_R(D) \neq G_F(D)$.

Since $z_{\text{core}}^{\mathcal{C}}$ only depends on the memory size of the encoder, so does the CWEF of each constituent code, $A^{\mathcal{C}}(2, U, Z)$ and, consequently, the union bound of all error events with information weight 2, $P(2)$. Therefore, the performance of a rate-1/3 PCCC, using a large interleaver size, mainly depends on the memory size of each constituent RSC encoder and not the underlying code, provided that the feedforward polynomial of each RSC encoder is different from the feedback primitive polynomial.

B. Punctured RSC Encoders

Rates higher than 1/2 can be achieved using a $2 \times M$ puncturing pattern $\mathbf{P}$ on a parent rate-1/2 RSC encoder $\mathcal{C}$. At
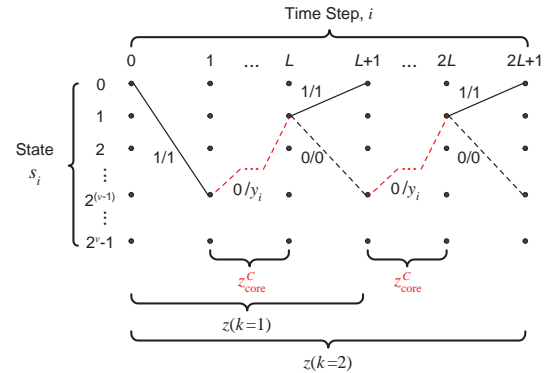

Fig. 2. Trellis diagram for codeword sequences of information weight two. Dashed lines correspond to paths generated when the rate-1/2 RSC encoder operates as a pseudo-random generator. A pair x_i/y_i next to a branch denotes the input and parity-check output bits, respectively, generated at the end of a state transition.

a time step i $(0 \leq i < N)$, the weights of the systematic and parity-check output bits of the punctured encoder $\mathcal{C}'$ will be $x_i \cdot p_{1,m}$ and $y_i \cdot p_{2,m}$, respectively, where x_i, y_i are the output bits of the parent rate-1/2 encoder and $p_{1,m}$, $p_{2,m}$ are the elements of column-m $(1 \leq m \leq M)$ of the puncturing pattern $\mathbf{P}$. Note that, owing to the systematic nature of the encoder, x_i also represents the input information bit. The relationship between m and i is

$$m = \text{rem}(i+1, M), \quad (15)$$

where $\text{rem}(i+1, M)$ denotes the remainder from the division $(i+1)/M$. Since the period of $\mathbf{P}$ is M, its elements are repeated in such a way that $p_{1,m} = p_{1,(m+jM)}$ and $p_{2,m} = p_{2,(m+jM)}$, where j is a non-negative integer.

In order to compute the CWEF of the punctured RSC for information weight $w = 2$, i.e., $A^{\mathcal{C}'}(2, U, Z)$, we need to derive an expression for the weight of the systematic and parity-check output sequences. Although information sequences with $w = 2$ generate paths of length $kL+1$, we first consider paths of length $L+1$, i.e., $k = 1$, for simplicity. The weight $u(k=1, m)$ of a systematic sequence, whose path diverges from the zero state when $p_{1,m}$ is active, is given by

$$u(k=1, m) = p_{1,m} + p_{1,(m+L)}, \quad (16)$$

since the two non-zero bits occur at the very beginning and at the very end of the path. Similarly, the weight $z(k=1, m)$ of the parity-check sequence, whose path diverges from the zero state when $p_{2,m}$ is active, assumes the form

$$z(k=1, m) = p_{2,m} + z_{\text{core}}^{m+1} + p_{2,(m+L)}, \quad (17)$$

since the parity-check bits at the beginning and at the end of the path are non-zero, while the weight of the remaining path is z_{core}^{m+1}, as it is illustrated in Fig.3. In order to calculate $z(k=1, m)$ for every value of m, we first need to derive $z_{\text{core}}^1, z_{\text{core}}^2, \ldots, z_{\text{core}}^M$, by applying the M circularly shifted versions of the puncturing vector $[p_{2,1}, \ldots, p_{2,M}]$ to the corresponding output parity-check bits of the parent rate-1/2 RSC encoder, i.e,

$$z_{\text{core}}^m = \sum_{i=1}^{L-1} \left(y_i \cdot p_{2,(i+m-1)} \right). \quad (18)$$

If we extend our analysis to codewords associated with paths of length $kL+1$, we obtain the generic expressions for $u(k, m)$ and $z(k, m)$ as follows

$$u(k, m) = p_{1,m} + p_{1,(m+kL)} \quad (19)$$

$$z(k, m) = p_{2,m} + \sum_{j=0}^{k-1} z_{\text{core}}^{m+jL+1} + p_{2,(m+kL)}, \quad (20)$$

where $z_{\text{core}}^{m+jM} = z_{\text{core}}^m$, due to the periodicity of the puncturing pattern. Since any codeword sequence, generated by an input sequence of weight 2, can be described by a polynomial $U^{u(k,m)} Z^{z(k,m)}$ for a given k and m, the summation of all polynomials of the form $U^{u(k,m)} Z^{z(k,m)}$ over all possible
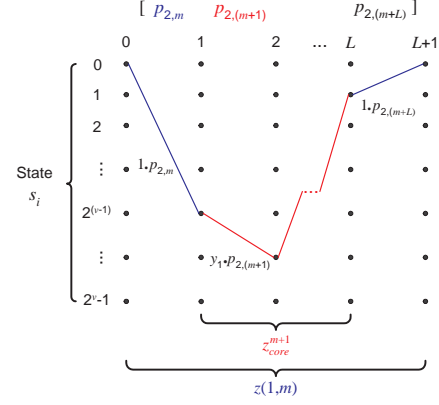

Fig. 3. Trellis diagram for the weight calculation of parity-check sequences
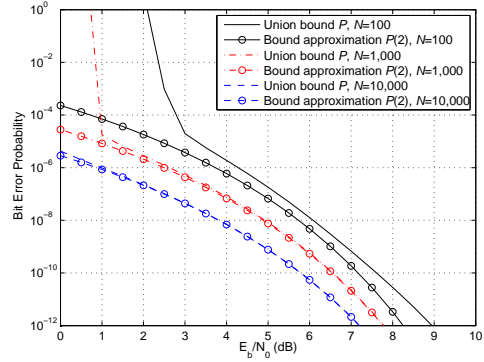

Fig. 4. Exact union bounds and their approximation for a rate-1/2 S-PCCC(1,17/15,17/15) using interleavers of various sizes

values of k and m will give the CWEF, $A^{\mathcal{C}'}(2, U, Z)$, of the punctured RSC code

$$A^{\mathcal{C}'}(2, U, Z) = \sum_{k=1}^{\lfloor (N-1)/L \rfloor} \sum_{m=1}^{M} A_{k,m}^{\mathcal{C}'} U^{u(k,m)} Z^{z(k,m)}, \quad (21)$$

where $A_{k,m}^{\mathcal{C}'}$ is the total number of codeword sequences with systematic weight $u(k, m)$ and parity-check weight $z(k, m)$. Coefficients $A_{k,m}^{\mathcal{C}'}$ can be easily derived if we observe that there are $N - kL$ codeword sequences of length $kL + 1$ each. The codeword sequences are grouped into M groups, whose members share the same weights $u(k, m)$ and $z(k, m)$. Thus, the number of codeword sequences in the m-th group is given by

$$A_{k,m}^{\mathcal{C}'} = \begin{cases} \left\lfloor \frac{N-kL}{M} \right\rfloor, & \text{if } \text{rem}\left((N-kL), M\right) < m \\ \left\lfloor \frac{N-kL}{M} \right\rfloor + 1, & \text{otherwise.} \end{cases} \quad (22)$$

Using (21), we can accurately and efficiently derive $P(2)$, i.e., the probability of all error events with information weight 2, which is a good approximation of the union bound P_b^u, for a large interleaver size. In the example shown in Fig.4, we see that $P(2)$ closely matches P_b^u, when the interleaver reaches the size of $N = 10,000$ bits.

IV. CASE STUDY: PSEUDO-RANDOM PUNCTURING

In this section we consider constituent RSC encoders employing primitive feedback polynomials, therefore the period L assumes the maximum value of $2^\nu - 1$. Furthermore, we assume that the elements of the puncturing vector $\mathbf{P}_Z$, for the parity-check output, form a pseudo-random sequence of period $M = L$, generated by the same primitive polynomial as that of the RSC encoder.

Since the puncturing period M is equal to the period L of the feedback polynomial, $u(k, m)$ and $z(k, m)$ are reduced to

$$u(k, m) = u(m) = 2p_{1,m}, \qquad (23)$$

$$z(k, m) = k z_{\text{core}}^{m+1} + 2p_{2,m}. \qquad (24)$$

Calculation of $z(k, m)$ and, consequently, $A^{\mathcal{C}'}(2, U, Z)$, requires knowledge of the L values of z_{core}^m. However, the assumption of pseudo-random puncturing can further simplify the computation of $z(k, m)$.

A. Derivation of the Minimum Weight Values

In order to express z_{core}^m in a more compact form, we first need to consider the autocorrelation function $\phi(j)$ of a polar sequence, which is defined as [13]

$$\phi(j) = \sum_{i=1}^{L} (2y_i - 1)(2y_{i+j} - 1) \qquad (25)$$

where $y_i = \{0, 1\}$ is the output of the parent rate-1/2 RSC encoder at time-step i for an input sequence of information weight 2, and $0 \leq j < L$. The parity-check sequence generated during the time period from $i = 1$ until $i = L$ is a pseudo-random sequence, provided that the encoder does not return to the zero state. In this case, the autocorrelation function can be reduced to [11], [14]

$$\phi(j) = \begin{cases} 2^\nu - 1, & \text{if } j = 0 \\ -1, & \text{if } 1 \leq j < L. \end{cases} \qquad (26)$$

Combining (25) and (26) we find that

$$\sum_{i=1}^{L} (y_i \cdot y_{i+j}) = \begin{cases} 2^{\nu-1}, & \text{if } j = 0 \\ 2^{\nu-2}, & \text{if } 1 \leq j < L. \end{cases} \qquad (27)$$

Since the puncturing vector for the parity-check bits, $\mathbf{P}_Z = [p_{2,i}]$, is also a pseudo-random sequence generated by the same primitive polynomial $G_R(D)$, such that $p_{2,i+1} = y_i$, we can rewrite expression (27) as follows

$$\sum_{i=1}^{L} (y_i \cdot p_{2,(i+m)}) = \begin{cases} 2^{\nu-1}, & \text{if } m = 1 \\ 2^{\nu-2}, & \text{if } 2 \leq m \leq L \end{cases} \qquad (28)$$

where j was replaced by m–1. Due to the structure of the RSC encoder, the last bit of the parity-check sequence is always zero, i.e., $y_L = 0$, therefore

$$\sum_{i=1}^{L-1} (y_i \cdot p_{2,(i+m)}) = z_{\text{core}}^{m+1} = \begin{cases} 2^{\nu-1}, & \text{if } m = 1 \\ 2^{\nu-2}, & \text{if } 2 \leq m < L \end{cases} \qquad (29)$$

so z_{core}^m is now a function of the memory ν of the RSC encoder. Having in mind that $p_{2,L+1} = 0$ since $y_L = 0$, and so is $p_{2,1}$,

and that $L = 2^\nu - 1$, the weight of a parity-check sequence assumes the form

$$z(k, m) = \begin{cases} k 2^{\nu-1}, & \text{if } m = 1 \\ k 2^{\nu-2} + 2p_{2,m}, & \text{if } 2 \leq m \leq 2^\nu - 1 \end{cases} \qquad (30)$$

where $2^{\nu-1}$ elements of the puncturing vector $\mathbf{P}_Z$ are equal to 1 and the remaining $2^{\nu-1} - 1$ are equal to 0, since the elements of $\mathbf{P}_Z$ form a pseudo-random sequence [11], [14].

The minimum weight of the parity-check sequences, $z_{\min}^{\mathcal{C}'}$ can be expressed as

$$z_{\min}^{\mathcal{C}'} = \min_{m=1\ldots L} \{z(k=1, m)\} = \begin{cases} 2, & \text{for } \nu = 2 \\ 2^{\nu-2}, & \text{for } \nu > 2 \end{cases} \qquad (31)$$

whereas the minimum weight of the codeword sequences, $d_{\min}^{\mathcal{C}'}$ assumes the form

$$d_{\min}^{\mathcal{C}'} = \min_{m=1\ldots L} \{u(m) + z(k=1, m)\}. \qquad (32)$$

As in the case of rate-1/3 PCCCs, we conclude that when a large interleaver is used, the performance of a PCCC whose parity-check sequences were punctured using pseudo-random patterns, mainly depends on the memory size of the constituent RSC encoders, and not the exact underlying codes.

B. Example Configurations for Rate-1/2 PCCCs

In order to maximize the minimum weight of the codeword sequences, $d_{\min}^{\mathcal{C}_1'}$, generated by the first constituent RSC encoder of a PCCC, we can set the puncturing vector for the systematic output, $\mathbf{P}_U = [p_{1,m}]$, to be the complement of the puncturing vector for the parity-check output $\mathbf{P}_Z = [p_{2,m}]$, i.e., $p_{1,m} = \overline{p_{2,m}}$. This configuration prevents $u(m)$ and $z(k, m)$ from assuming the smallest values at the same time. Therefore, expression (32) becomes

$$d_{\min}^{\mathcal{C}_1'} = 2 + 2^{\nu-2}, \qquad (33)$$

for $\nu \geq 2$.

A code rate of 1/2 can be achieved, if the parity-check output of the second RSC encoder is not punctured. In that case, $z_{\min}^{\mathcal{C}_2'}$ can be derived from (11) and (14). The free effective distance of the corresponding PS-PCCC assumes the form

$$d_{\text{free,eff}} = 4 + 3(2^{\nu-2}). \qquad (34)$$

We refer to this example configuration as "Pseudo A".

If our objective is to obtain a turbo code whose BEP performance quickly converges to the union bound but experiences a high error floor, we need to increase the number of transmitted systematic bits [10], [15], [16]. The parity-check output of both the first and the second constituent encoder is punctured using the same vector $\mathbf{P}_Z$. Bearing in mind that $\mathbf{P}_U$ is taken to be the complement of $\mathbf{P}_Z$, we need to replace all but one of the 0's in $\mathbf{P}_U$ with 1's, in order to achieve a code rate of 1/2. The minimum codeword weight $d_{\min}^{\mathcal{C}_1'}$ for the first constituent encoder is given by (33), while the minimum parity-check weight $z_{\min}^{\mathcal{C}_2'}$ for the second constituent encoder is

TABLE I
PUNCTURING PATTERNS FOR RATE-1/2 PCCC(1,17/15,17/15)

	Pseudo A	Pseudo B	Litt A	Litt B
Vector for Sys.Output	[1000101]	[1111101]	[0010]	[11]
Vector for 1st Par.Output	[0111010]	[0111010]	[1101]	[10]
Vector for 2nd Par.Output	[1111111]	[0111010]	[1111]	[01]

given by (31). The summation of the two minimum weights yields the free effective distance of the PS-PCCC

$$d_{\text{free,eff}} = \begin{cases} 5, & \text{for } \nu = 2 \\ 2 + 2^{\nu-1}, & \text{for } \nu > 2. \end{cases} \quad (35)$$

We refer to this example configuration as "Pseudo B".

The particular puncturing patterns of each example configuration for the case of PCCC(1, 17/15, 17/15) are presented in Table I. The configuration denoted as "Litt A" achieves a very low error floor and it was obtained through exhaustive search using [10], whereas "Litt B" is the conventional approach for obtaining rate-1/2 turbo codes.

C. The Benefits of Pseudo-random Patterns

Good punctured PCCCs can only be found by means of an exhaustive search among all possible patterns of a specific puncturing period M. The selection of a good pattern is not intuitive, since it can lead to catastrophic puncturing [15], i.e., $d_{\min}^{\mathcal{C}'} = 0$, or semi-catastrophic puncturing, i.e., $z_{\text{core}}^m = 0$ for some values of m, of a constituent code $\mathcal{C}'$. Furthermore, calculation of $d_{\min}^{\mathcal{C}'}$ and $z_{\min}^{\mathcal{C}'}$ requires prior knowledge of the M values of z_{core}^m.

The selection of a pseudo-random puncturing pattern guarantees that $z_{\text{core}}^m > 0$, and consequently, $d_{\min}^{\mathcal{C}'} > 0$. Moreover, z_{core}^m can be expressed as a function of the memory size ν of $\mathcal{C}'$, permitting the immediate derivation of the minimum weights that characterize the PCCC. For the given puncturing rate of the parity-check output, the minimum value of z_{core}^m is maximised, and so is $z_{\min}^{\mathcal{C}'}$, due to the properties of pseudo-random sequences.

In Fig.5 we have plotted the performance of all four rate-1/2 PCCC(1, 17/15, 17/15) configurations, presented in Table I. We observe that "Pseudo B" slightly outperforms the conventional "Litt B" configuration, while the performance of the PCCC based on the easy to derive "Pseudo A" pattern is close to the performance of the PCCC based on the "Litt A" pattern, obtained through exhaustive search.

V. CONCLUSIONS

We presented a simple approach to calculate the CWEF of punctured RSC codes, for input sequences of minimum information weight, which facilitates the approximation of the upper bound to the BEP, for punctured PCCCs using large interleaver sizes. Our technique offers the advantage of simplicity and reduced complexity, compared to time-hungry approaches, such as brute-force, or the more complex methods developed in [7], [10].

Furthermore, we considered pseudo-random puncturing patterns as a case study for our technique and we demonstrated
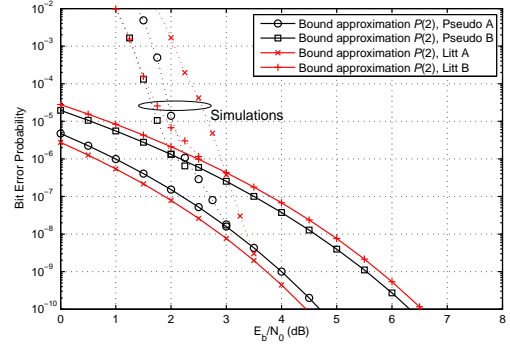

Fig. 5.   Bounds for various rate-1/2 PCCC(1,17/15,17/15) configurations, using an interleaver of size 1,000 bits

that they prevent catastrophic or semi-catastrophic puncturing and facilitate the calculation of the minimum output weights of a turbo encoder, which characterize the performance of PCCCs. We concluded that pseudo-random puncturing could be used to obtain rate-1/2 PCCCs exhibiting low error floors, while specific puncturing patterns that achieve either a lower error floor or quicker convergence to the ML performance bound, could be determined by a subsequent search.

REFERENCES

[1] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1261–1271, Oct. 1996.
[2] D. Divsalar, S. Dolinar, R. J. McEliece, and F. Pollara, "Transfer function bounds on the performance of turbo codes," JPL, Cal. Tech., TDA Progr. Rep. 42-121, Aug. 1995.
[3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 409–429, Mar. 1996.
[4] ——, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, no. 5, pp. 591–600, May 1996.
[5] Ö. Açikel and W. E. Ryan, "Punctured turbo-codes for BPSK/QPSK channels," *IEEE Trans. Commun.*, vol. 47, pp. 1315–1323, Sept. 1999.
[6] F. Babich, G. Montorsi, and F. Vatta, "Design of rate-compatible punctured turbo (RCPT) codes," in *Proc. ICC'02*, N.Y., USA, Apr. 2002, pp. 1701–1705.
[7] M. A. Kousa and A. H. Mugaibel, "Puncturing effects on turbo codes," *Proc. IEE Comm.*, vol. 149, pp. 132–138, June 2002.
[8] M. Fan, S. C. Kwatra, and K. Junghwan, "Analysis of puncturing pattern for high rate turbo codes," in *Proc. MILCOM'99*, USA, Oct. 1999, pp. 547–500.
[9] I. Land and P. Hoeher, "Partially systematic rate 1/2 turbo codes," in *Proc. Int. Symp. Turbo Codes*, France, Sept. 2000, pp. 287–290.
[10] I. Chatzigeorgiou, M. R. D. Rodrigues, I. J. Wassell, and R. Carrasco, "A novel technique for the evaluation of the transfer function of punctured turbo codes," in *Proc. ICC'06*, Turkey, July 2006.
[11] F. J. MacWilliams and N. J. A. Sloane, "Pseudo-random sequences and arrays," *Proc. IEEE*, vol. 64, pp. 1715–1729, Dec. 1976.
[12] L. C. Perez, J. Seghers, and D. J. Costello, "A distance spectrum interpretation of turbo codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1698–1709, Nov. 1996.
[13] J. G. Proakis, *Digital Communications*, 4th ed.   McGraw-Hill, 2001.
[14] D. V. Sarwate and M. B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proc. IEEE*, vol. 68, pp. 593–618, May 1980.
[15] S. Crozier, P. Guinand, and A. Hunt, "On designing turbo-codes with data puncturing," in *Proc. Can. Workshop Inf. Theory*, Canada, 2005.
[16] Z. Blazek, V. K. Bhargava, and T. A. Gulliver, "Some results on partially systematic turbo codes," in *Proc. VTC'02-Fall*, Canada, Sept. 2002, pp. 981–984.